**qsub** *[<options>] <executable> [<executable options>]*

# DESCRIPTION

**qsub** submits a job to the queue manager for execution.

**JOBID EXPANSION ($jobid or $COBALT_JOBID):**

If $jobid or $COBALT_JOBID is specified in any jobid expansion option then it will be converted to the actual jobid.

Example: qsub ... --env myenv=\$jobid_myval ...

If the jobid is 123 then myenv will be set to '123_myval'

jobid expansion options: --env, --jobname, -o, -e, -O, --debuglog

Note: Remember to escape appropriately "\$jobid" or "\$COBALT_JOBID"

**SCRIPT JOB DIRECTIVES (#COBALT):**

Command options can be entered in a script job by using #COBALT <options>. You can enter all in one directive or directives. The #COBALT tag should be right after the first line of the script. qsub will continue to read directives until it reads the first line that does not contain the tag #COBALT.

The command line options will take precedence over the Cobalt script jobx directives.

Note: An error will occur if in the script the --mode option is speci? fied and it is not "script"

Example:

```
  script.sh:

    #!bin/bash
    #COBALT -t 50 -n 30
    echo "SCRIPT JOB WITH DIRECTIVE"

  or

    #!bin/bash
    #COBALT -t 50
    #COBALT -n 30
    echo "SCRIPT JOB WITH DIRECTIVE"

  qsub script.sh sargs
  is equivalent to
  qsub -t 50 -n 30 --mode script script.sh sargs

  qsub -t 75 script.sh sargs
```

**--version**
Displays client revision and Cobalt version

**-A --project [project]**

Associate the job with the allocation for project project. This is used to properly account for machine usage.

**--attrs [<attr1=val1>:<attr2=val2>:...]**

Set a list of attributes for a job that must be fulfilled for a job to run. For instance, --attrs location=[blocklocation] will force a job to run on a specific block.

**--cwd [directory]**

Tell the job to use the specified directory as the default cur? rent working directory while running the job.

**--debuglog [path to file] - (jobid expansion)**

Send debugging information from cobalt to file specified.

**--dependencies [<jobid1>:<jobid2>:...]**

Set the dependencies for the job being submitted. This job won't run until all jobs in the dependency list have finished and exited with a status of 0.

**--disable_preboot**

If --mode script is specified, this will cause Cobalt to run the job script without booting the block first. By default the script is executed after the requested block is booted.

**-e --error [path to file] - (jobid expansion)**

Send job stderr to file specified.

**--env [<var1=value1>:<var=value2>] - (jobid expansion)**

Set environment variables that will be passed into the job's environment. If the the characters ":" and "=" are within the value, right hand side (rhs) of a environment variable assign? ment, then need to quote the entire string and precede any ":" and "=" characters that are in the rhs with "\". runtime.

If multiple --env options are specified then qsub will take the union.

e.g. qsub ... --env "A=B:C=x\=1\:y\=2\:z\=3:D=ABC" ... --env "MY_NAME=cobalt' ...

A would equal B

C would equal x=1:y=2:z=3

D would equal ABC

and
Submit job and immediately place it in the user hold state

**-i --inputfile [path to file]**

Send file to job's stdin.

**-I (NOT IMPLEMENTED)**

Run an interactive command. During job execution, the current tty will be connected with the running job. (Not available on Blue Gene systems.)

**-M --notify [email address]**

Send an email notification at the start and stop of the job to the specified email address.

**-n --nodecount [nodecount]**

Specifies the node count for a job

**-o --output [path to file] - (jobid expansion)**

Send job stdout to file specified.

**-O --outputprefix [output prefix] - (jobid expansion)**

Use the specified prefix for both .output, .error and debuglog files. This will only take effect if -o or -e or --debuglog not supplied.

**--proccount [process count]**

Specify the number of processes to start.

**-q --queue [queue]**

Submit the job to the specified queue.

**--run_users --user_list [<user1>:<user2>:...]**

Sets a colon-separated list of users for the job being submit? ted. All users in this list will be able to execute cobalt com? mands to control the job. The submitting user is always able to run commands on a submitted job.

**--run_project**

Appends the list of users in a project to the list of users authorized to run commands on the submitted job.

**-u --umask**

set umask: octal number default(022)

**-t --time [wallclock time]**

Specify the runtime for a job. If the job runs over this limit, it will be killed. The time may be specified as eitehr an inte? ger number of minutes or a colon-delimited value of the format: HH:MM:SS. Enter 0 to get the max allowed walltime.

Run the job with the specified IO Node kernel profile. (Blue? Gene/Q only)

**--ion_kerneloptions [kernel options]**

Run the job with the specified IO Node kernel options. (Blue? Gene/Q only)

**--mode [mode]**

This is a platform-dependent setting that controls the execution mode of the submitted job and the division of cores and memory on the compute nodes. Valid values are dependent on the spe? cific BlueGene platform:

BG/L: Valid values are "co" (coprocessor),"vn" (virtual node) and "script".
The default is co.

BG/P: Valid values are "smp", "dual", "vn" and "script". "smp", "dual" and
"vn" correspond to the modes within mpirun. The default is "smp"

BG/Q: Valid values are "c1","c2","c4","c8","c16","c32","c64", and "script".
The c(n) modes correspond to the "ranks-per-node" flag of runjob and
governs the MPI ranks started per node on the compute nodes. Node memory
is divided evenly between all processes. n/64 threads will be available
to each MPI process as well. If this option is not set the default is "c1"

On all platforms, the "script" mode causes the job to be routed through the
script-forker, and will allow the execution of script jobs allows the user to

# NOTE

The only thing printed to STDOUT is the jobid, any other error or informational messages are printed to STDERR.

# SEE ALSO

**qstat(1)**, **qdel(1)**, **qhold(1)**, **qrls(1)**, **qalter(1)**, **qmove(1)**, **cqm(1)**, **cqwait(1)**

# BUGS

**qsub(1)**